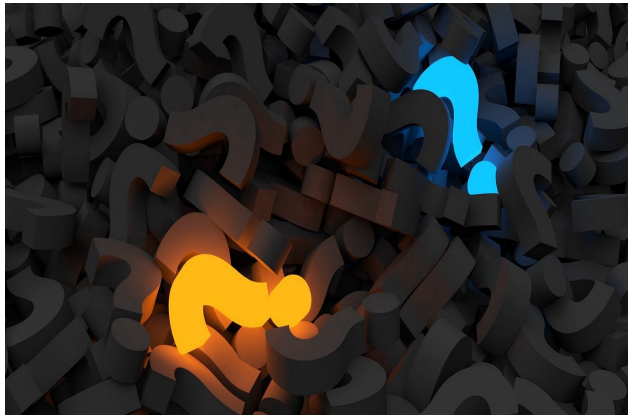


Quora Question Pair Similarity

Group 36

Dao Ngoc Hieu

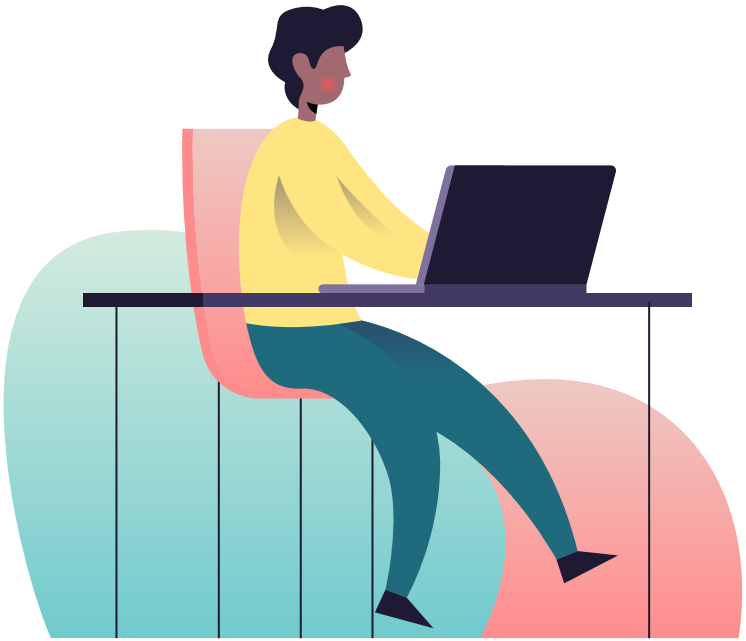
Ho Kah Hsin Carine

Juliet Teoh Qian Ying

Noorus Suhaina

Olivia Juliani Johansen

CONTENTS



01

INTRODUCTION

BACKGROUND

Quora

- Question Answering website
- Duplicate questions on these sites are very common
- Duplicate questions may prevent a user from seeing high quality response that already exists
- Enhance user experience by identifying duplicates

DATASET

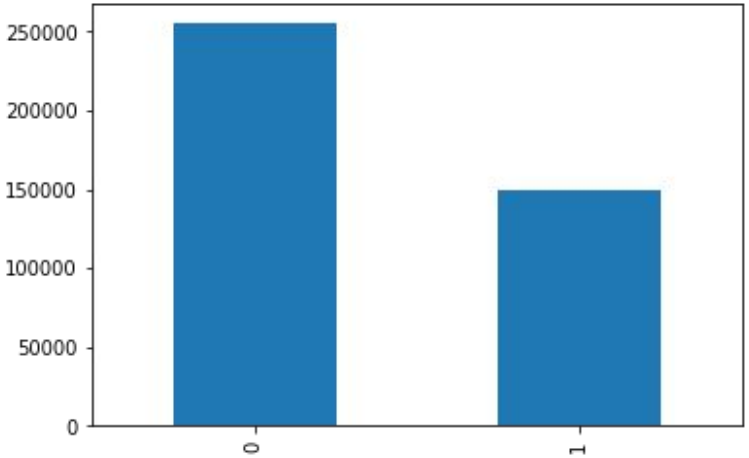
id	qid1	qid2	question1	question2	is_duplicate
0	1	2	What is the step by step guide to invest in share market in india?	What is the step by step guide to invest in share market	0

- id - the id of a training set question pair
- qid1, qid2 - unique ids of each question (only available in train.csv)
- question1, question2 - the full text of each question
- is_duplicate - the target variable, set to 1 if duplicates and 0 otherwise

02

DATA PRE-PROCESSING

IMBALANCED DATASET



Imbalanced dataset

- Misleading interpretation of accuracy
- 404290 training instances where 149,263 instances are duplicates while 255,027 of them are non-duplicates.

Generated data

A and B duplicates

B and C duplicates

A and C duplicates

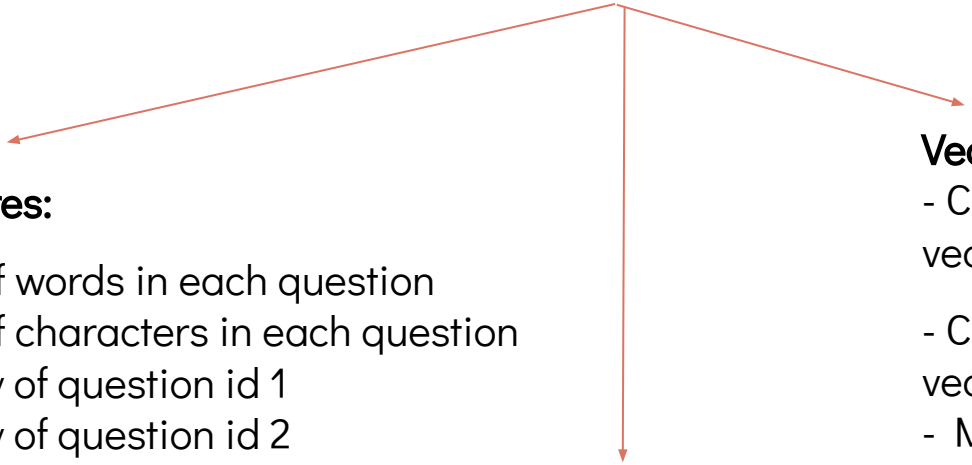
Cleaning text

- Lower sentence case
- Converted numerical values, special symbols
- Decontract words (“don’t becomes do not”)
- Stop words removal
- Lemmatization

3 versions of data cleaning

1. Text cleaning *“cleaned_features”*
2. Text cleaning + lemmatization
3. Text cleaning + lemmatization + stop words removal *“stopwords_lemmatize”*

FEATURE EXTRACTION



Basic features:

- Number of words in each question
- Number of characters in each question
- Frequency of question id 1
- Frequency of question id 2
- Total unique number of words in question 1 and 2
- Total words
- Common words count

Advanced Features:

- Simple Ratio
- Partial Ratio
- Token Sort Ratio
- Token Set Ratio

Using fuzzywuzzy package

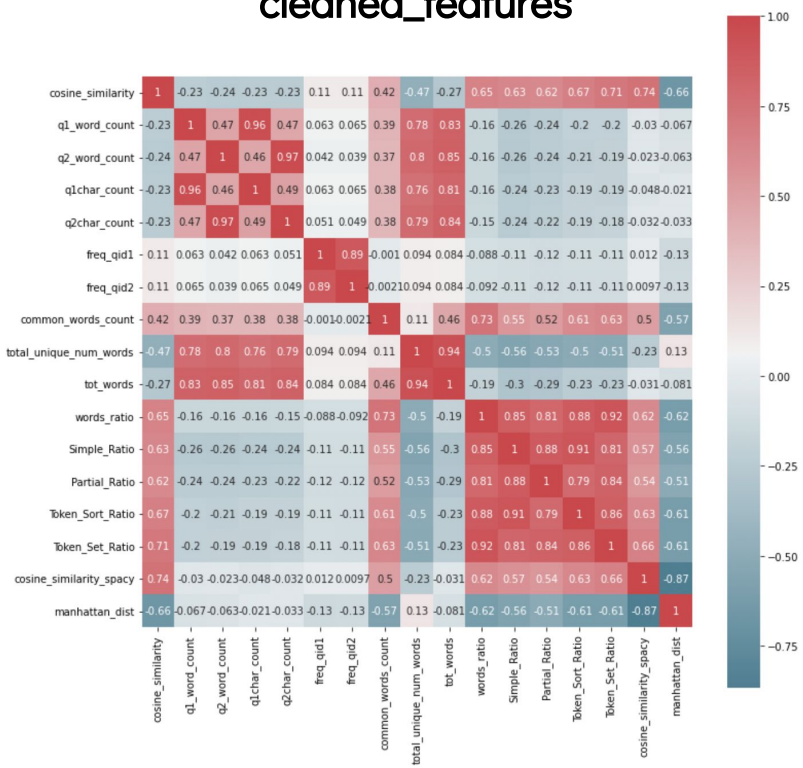
Vector features:

- Cosine similarity between sentence vectors encoded with Sentence-Bert

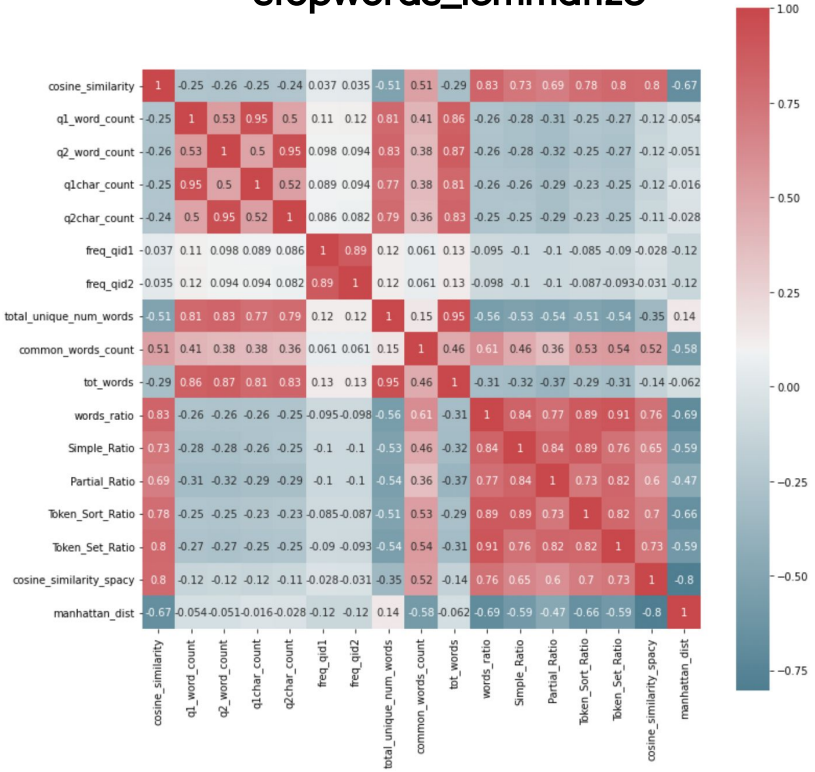
- Cosine similarity between mean word vectors encoded with Spacy
- Manhattan distance between mean word vectors encoded with Spacy

CORRELATION MATRIX

cleaned_features



stopwords_lemmatize



FEATURE SELECTION

Remove highly correlated features (used threshold > 0.9)

			cleaned_features	stopwords_lemmatize
No.	Feature 1	Feature 2	Correlation coefficient > 0.9	
1	q1_word_count	q1_char_count	0.96	0.95
2	q2_word_count	q2_char_count	0.97	0.95
3	tot_words	total_unique_num_words	0.94	0.95
4	simple_ratio	token_sort_ratio	0.91	-
5	words_ratio	token_sort_ratio	-	0.91

FEATURE SELECTION

Remove bias features: freq_qid1, freq_qid2

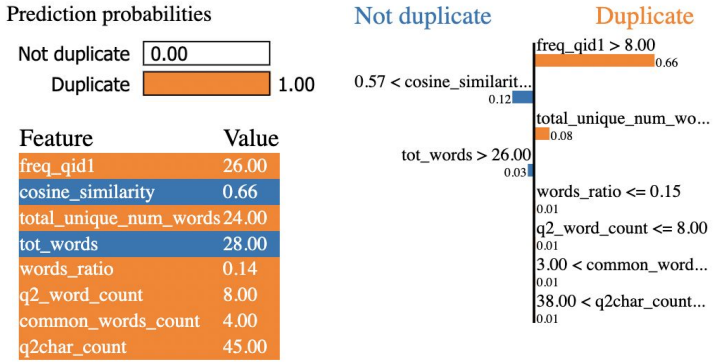
- freq_qid1: number of times the question identified by qid1 occurs
- freq_qid2: number of times the question identified by qid2 occurs
- Data generation: is_duplicate(q1, q2) == 1 and is_duplicate(q2, q3) == 1 then add is_duplicate(q1, q3) == 1

FEATURE SELECTION

Remove bias features: freq_qid1, freq_qid2

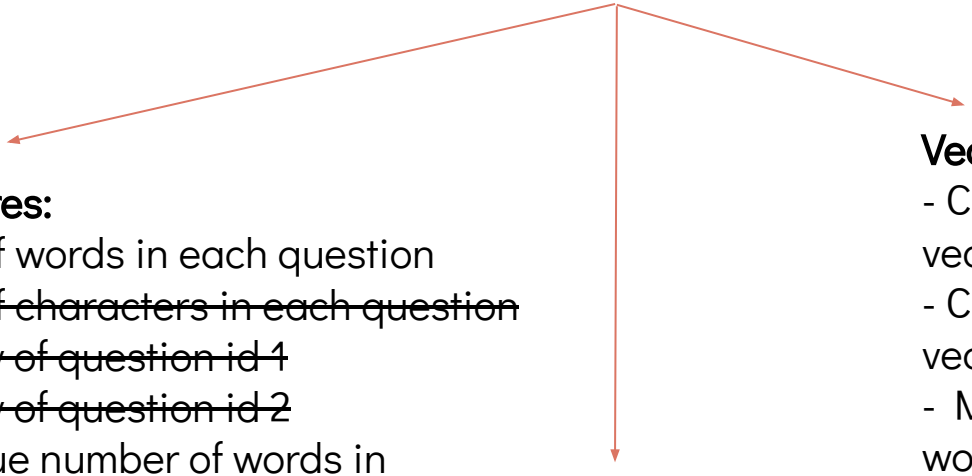
- For example:

Q1: “why do people waste time waiting for answers on quora rather than google a question and get an instant answer”
Q2: “why do not people just google their questions”



	question1_cleaned	question2_cleaned
49102	why do people waste time waiting for answers o...	why do so many people prefer to ask questions ...
67668	why do people waste time waiting for answers o...	why do people write questions on quora that co...
81493	why do people waste time waiting for answers o...	why do quorans ask questions for which authent...
83752	why do people waste time waiting for answers o...	why do most quorans ask questions here instead...
95724	why do people waste time waiting for answers o...	why do some people ask simple direct science q...
120905	why do people waste time waiting for answers o...	why do so many people ask questions on quora c...

FINAL FEATURE SET



Basic features:

- Number of words in each question
- ~~Number of characters in each question~~
- ~~Frequency of question id 1~~
- ~~Frequency of question id 2~~
- Total unique number of words in question 1 and 2
- ~~Total words~~
- Common words count

Advanced Features

- Simple Ratio
- Partial Ratio
- ~~Token Sort Ratio~~
- Token Set Ratio

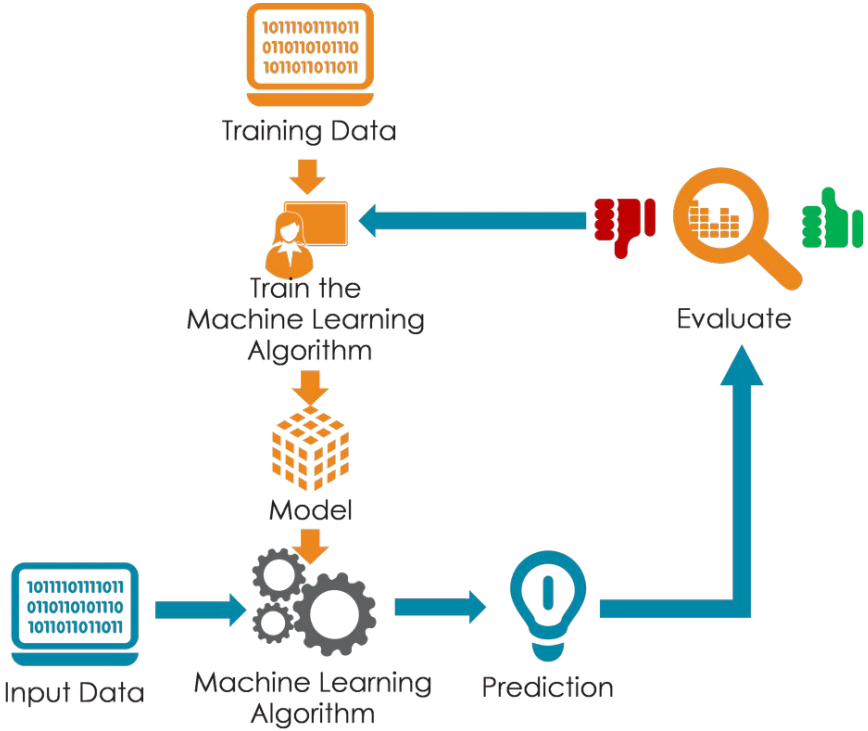
Using fuzzywuzzy package

Vector features:

- Cosine similarity between sentence vectors encoded with Sentence-Bert
- Cosine similarity between mean word vectors encoded with Spacy
- Manhattan distance between mean word vectors encoded with Spacy

03

MODELING


Training Data
Train the Machine Learning Algorithm
Evaluate
Model
Input Data
Machine Learning Algorithm
Prediction

MODELING

1

Logistic Regression

Baseline model

2

LSTM Model

RNN architecture to reduce vanishing gradient problem

3

Random Forest

Ensemble learning method

MODEL INPUTS

1

Logistic Regression

Word vectors from spacy, feature set

2

LSTM Model

Glove word vectors

3

Random Forest

Feature set

LOGISTIC REGRESSION

- Logistic regression is a statistical analysis method used to predict a data value based on prior observations of a data set.
- Implementation
  - sklearn logistic regression
  - GridSearchCV to find optimal parameters:
    - L2 regularization is applied with C < 1 where hyperparameter C is the inverse of regularization strength. L2 is preferred over L1 as there are **many variables with small/medium effect** that L1 might eliminate.
    - Solver used to find parameters is lbfgs. Lbfgs is preferred over sag as it is **more robust to unscaled data**

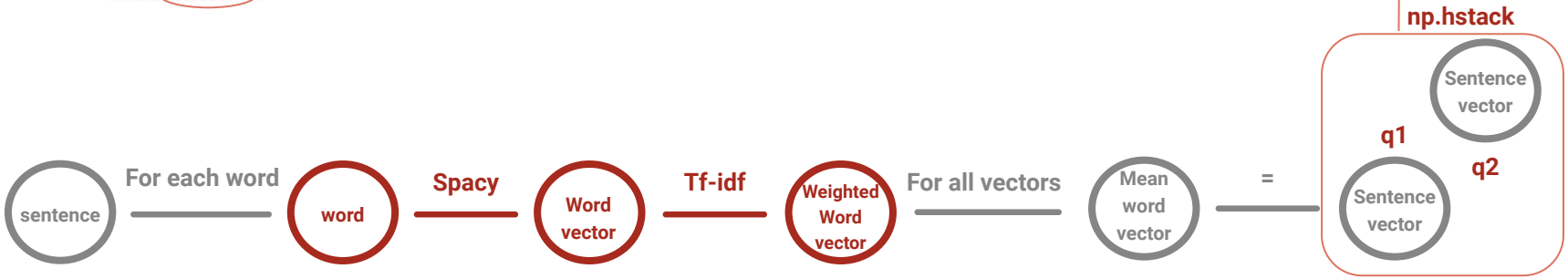
Word Vectors with Spacy

- Each word is vectorised using the Spacy library
- We take tf-idf weighted word vectors because not all words equally represent a sentence
- To format the input into logistic regression model, we took the mean vector of all the words in a sentence to represent the vector of the sentence.
    - According to Kenter et al. 2016, "simply averaging word embeddings of all words in a text has proven to be a strong baseline or feature across a multitude of tasks", such as short text similarity tasks.

Model Input

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
420840	-1.036242	1.517745	-1.883154	-0.915116	0.470286	0.121525	-0.106615	-0.461391	-0.523461	15.597747	-0.349245	0.611440	0.194022	1.060797
165062	-2.266786	-0.082580	0.045319	-0.360821	0.211304	0.364702	-1.009413	-2.384191	0.541124	12.382317	-3.932239	1.891618	-0.361282	1.378702
322109	0.514605	0.906266	-0.035650	1.170772	0.465645	1.883955	1.207154	-1.728961	0.735067	13.862038	-0.702870	1.947318	-0.960853	0.307402
87324	-0.256014	1.285137	-0.494681	-0.295548	1.560920	-1.252197	0.252921	-1.177463	0.420894	8.402436	-1.875473	-0.069674	-1.188863	-1.054325
236228	-1.988174	0.791863	-1.410799	0.234462	-0.001003	-1.851774	-0.093459	-1.909775	3.157229	16.139422	-1.210072	-0.933962	-0.028359	-0.287306

5 rows × 300 columns



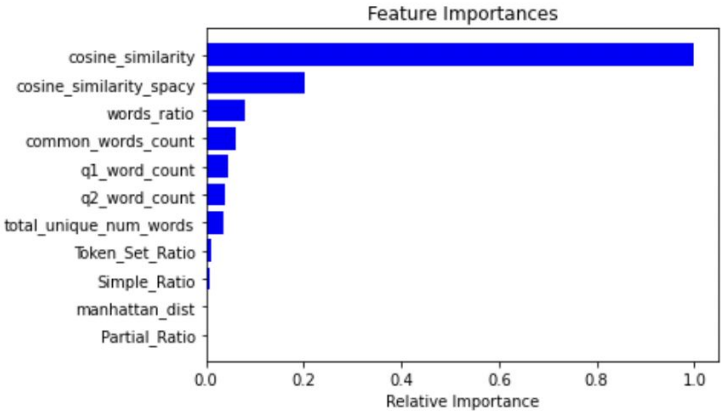
LOGISTIC REGRESSION

Training on horizontally stacked q1 vector, q2 vector

cleaned_features	stopwords_lemmatize
Accuracy: 0.71075 Precision: 0.71 Recall: 0.72 F1 Score: 0.71074 AUC-ROC: 0.710793 Log Loss: 0.570770 RMSE: 0.53781	Accuracy: 0.697775 Precision: 0.69 Recall: 0.71 F1 Score: 0.69777 AUC-ROC: 0.69780 Log Loss: 0.58644 RMSE: 0.54974
hyperparameters	
Regularization: L2 C: 0.0001 Solver: lbfgs	

- cleaned_features and stopwords_lemmatize **perform similarly** when trained on sentence vectors
- Noise from stopwords and un-lemmatized words offset additional information they provide

FEATURE IMPORTANCE

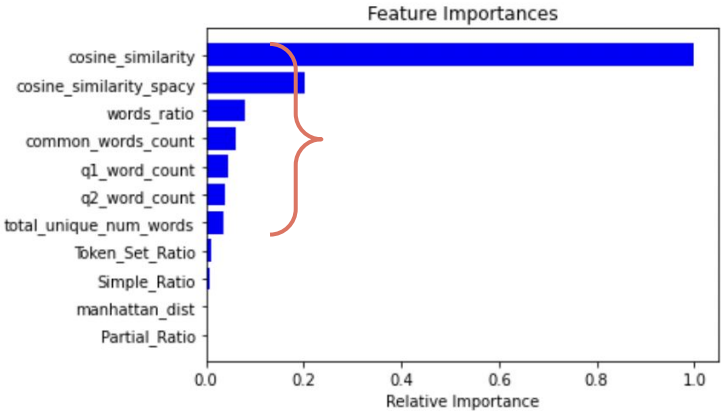


cleaned_features

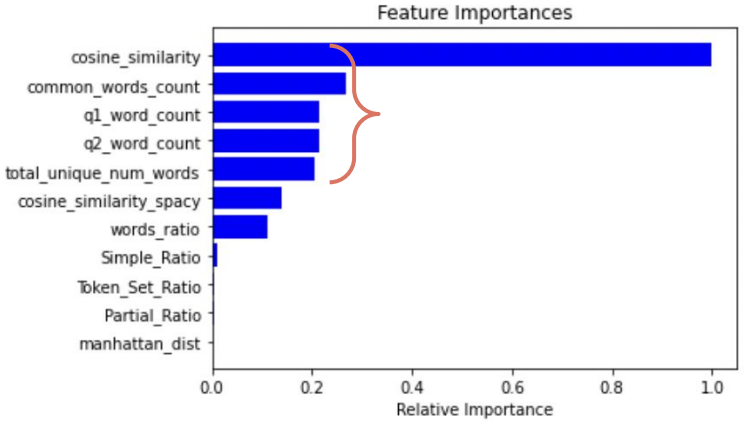
stopwords_lemmatize

- Common patterns: cosine_similarity is the major deciding factor for logistic regression
- Features in stopwords_lemmatize have higher absolute importance than features in cleaned_features
- cosine_similarity more important than cosine_similarity_spacy
- Word counts more important for stopwords_lemmatize

FEATURE SELECTION



cleaned_features



stopwords_lemmatize

Choose features with relatively high importance to reduce chance of overfitting

- cleaned_features:
  { cosine_similarity, cosine_similarity_spacy, words_ratio, common_words_count, q1_word_count, q2_word_count, total_unique_num_words }

- stopwords_lemmatize:
  { cosine_similarity, common_words_count, q1_word_count, q2_word_count, total_unique_num_words }

LOGISTIC REGRESSION

Training on features

cleaned_features	stopwords_lemmatize
Accuracy: 0.76145 Precision: 0.73 Recall: 0.83 F1 Score: 0.76050 AUC-ROC: 0.76189 Log Loss: 0.48362 RMSE: 0.48841	Accuracy: 0.6869 Precision: 0.66 Recall: 0.74 F1 Score: 0.68591 AUC-ROC: 0.68729 Log Loss: 0.57597 RMSE: 0.55955
hyperparameters	
Regularization: L2 C: 0.368 Solver: lbfgs	

- cleaned_features gives **better metrics** compared to stopwords_lemmatize
- Removing stop words or lemmatization may have introduced noises, and thus performances of it is not as good as simply the cleaned features

LOGISTIC REGRESSION

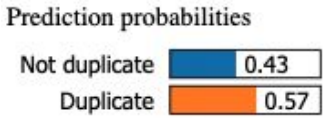
Incorrect prediction: false positive

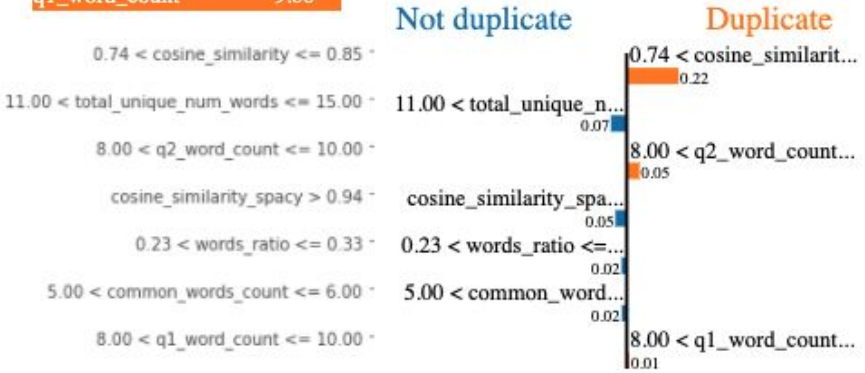
Actual: Not duplicate
Predicted: Duplicate

cleaned_features

Question 1: "how can i prepare for exam before 2 days"
Question 2: "how can i prepare for exams in 5 days"

Feature	Value
cosine_similarity	0.81
total_unique_num_words	12.00
q2_word_count	9.00
cosine_similarity_spacy	0.97
words_ratio	0.33
common_words_count	6.00
q1_word_count	9.00

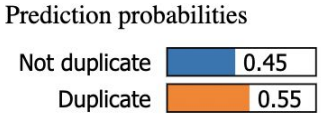


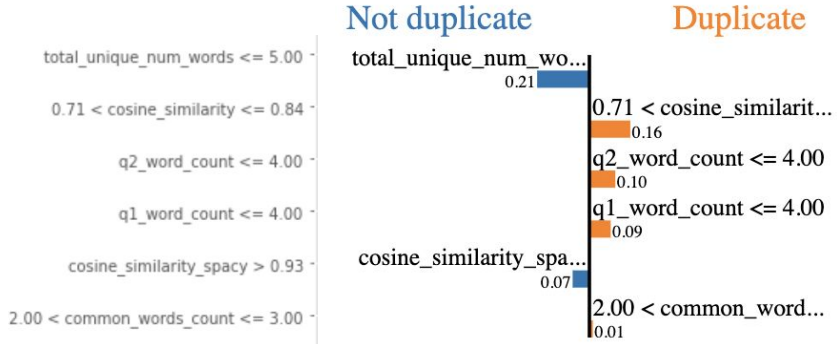


stopwords_lemmatize

Question 1: "prepare two day exam"
Question 2: "prepare exam day"

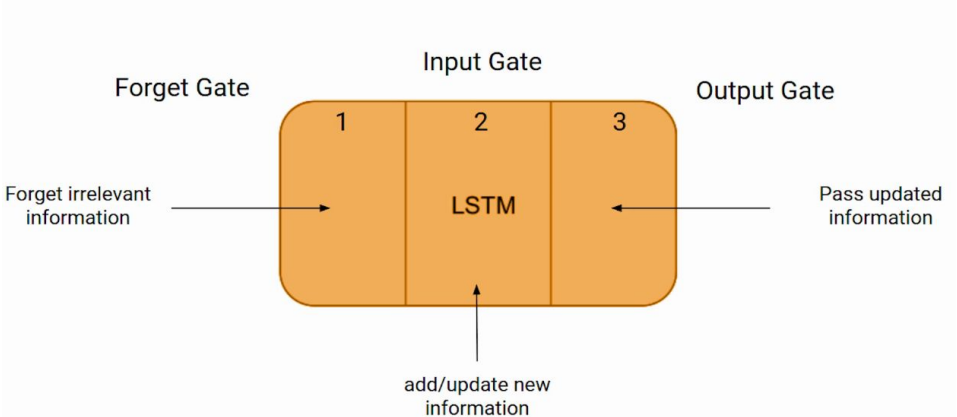
Feature	Value
total_unique_num_words	4.00
cosine_similarity	0.79
q2_word_count	3.00
q1_word_count	4.00
cosine_similarity_spacy	0.96
common_words_count	3.00





LSTM

- Motivation: Fully utilize the information provided through word embeddings
  - LSTM is a type of RNN and RNNs allow word embeddings of each word to be sequentially inputted into the model to get an output.
  - An LSTM cell is composed of an input gate, an output gate and a forget gate and this helps LSTMs regulate the past data that it retains and gather context from the past data.
  - We hypothesise that this would make the LSTM better at understanding sentences and perform better than the logistic regression models.



"What is the GDP of America?"

Tokenization of text

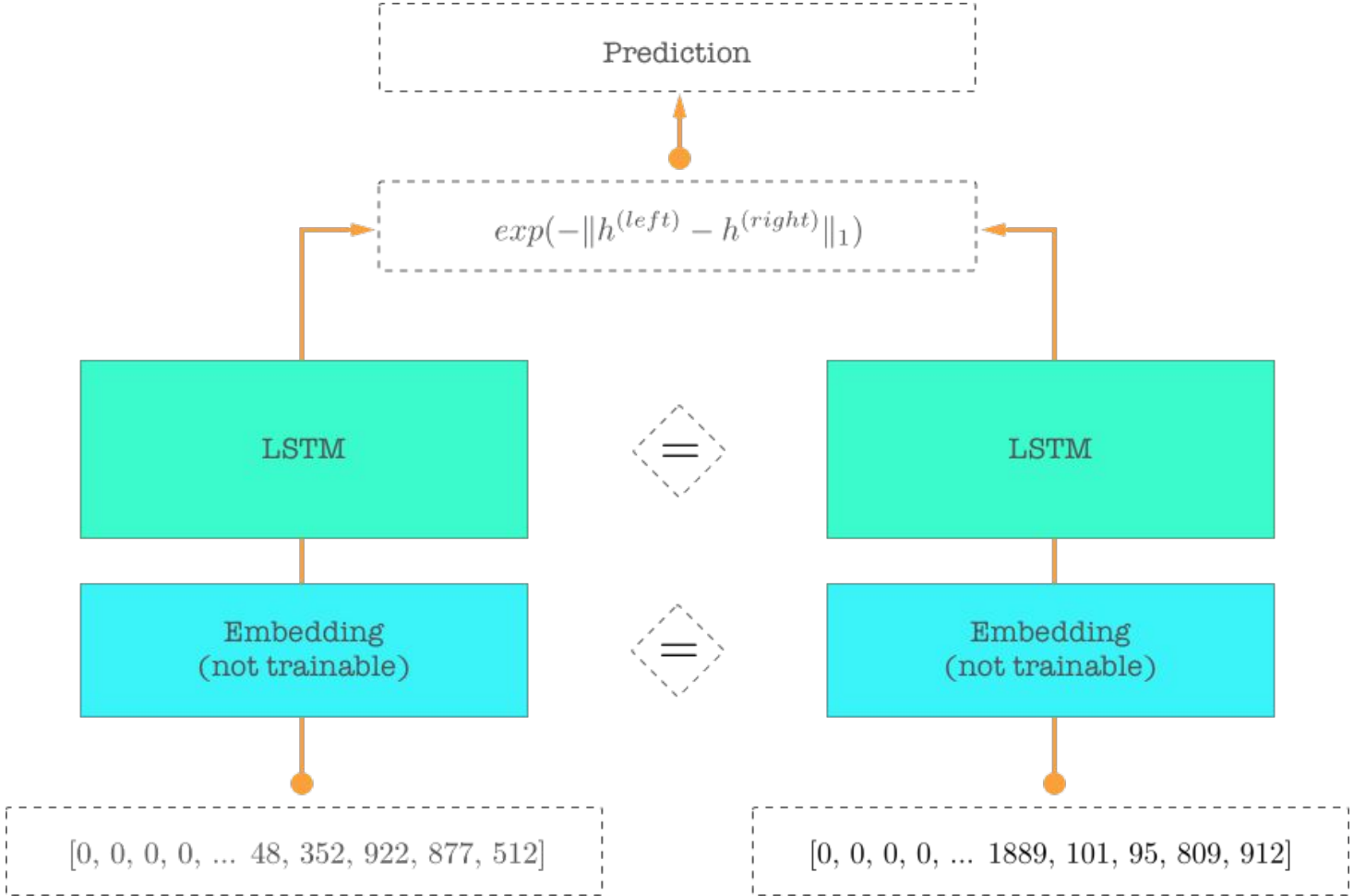
[29, 35, 912, 461 .... 0, 0, 0]

Embedding Lookup in Token to Word Vector Matrix

$$\begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0.21 & 2.4 & \dots & \dots \\ 1.02 & -1.2 & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ -0.4 & -0.06 & \dots & \dots \end{bmatrix}$$

Matrix representing the question, where each vector column is the embedding of the corresponding word

maLSTM


Prediction
$exp(-\|h^{(left)} - h^{(right)}\|_1)$
LSTM
LSTM
=
Embedding
(not trainable)
Embedding
(not trainable)
=
[0, 0, 0, 0, ... 48, 352, 922, 877, 512]
[0, 0, 0, 0, ... 1889, 101, 95, 809, 912]

Other LSTM experiments

- We decided to add some dense layers to our model and let it find the similarity between the 2 LSTM outputs through deep learning

    - Concatenate the LSTM outputs into a vector before passing it through some dense layers and ending with a sigmoid activation function
        - Allows the model to find its own distance function by itself through the dense layers

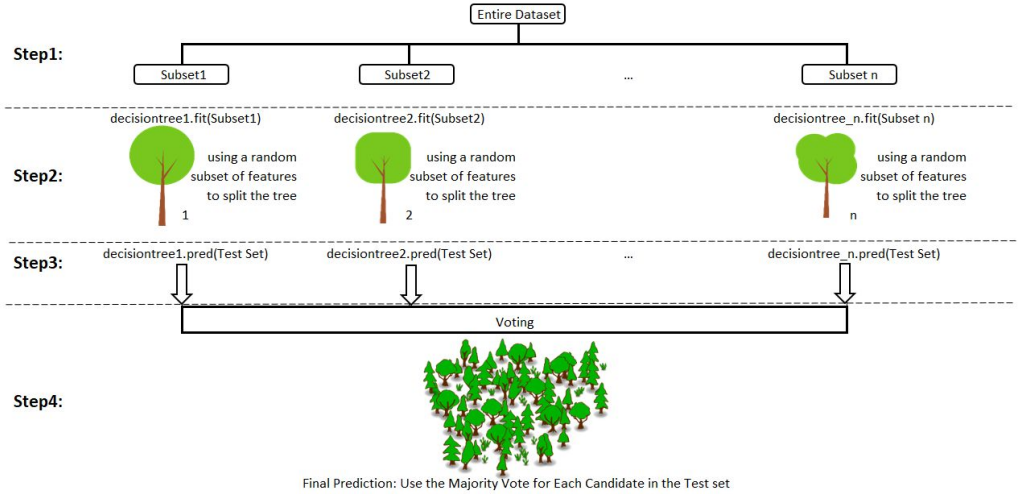
    - Use a Hadamard Product on the LSTM outputs before passing it through some dense layers and ending with a sigmoid activation function.
        - Inspired from cosine similarity
        - Multiplying these outputs could help the model determine if the sentences were similar or not.

LSTM results

maLSTM	Accuracy: 0.849098 Precision: 0.873697 Recall: 0.814733 F1 Score: 0.838711 AUC-ROC: 0.925007 Log Loss: 0.108170
LSTM - concatenate lstm outputs	Accuracy: 0.847156 Precision: 0.830691 Recall: 0.870764 F1 Score: 0.846148 AUC-ROC: 0.929831 Log Loss: 0.380886
LSTM - multiply lstm outputs	Accuracy: 0.852942 Precision: 0.834753 Recall: 0.879597 F1 Score: 0.852591 AUC-ROC: 0.935025 Log Loss: 0.356903

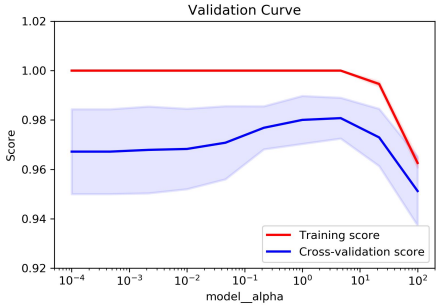
RANDOM FOREST

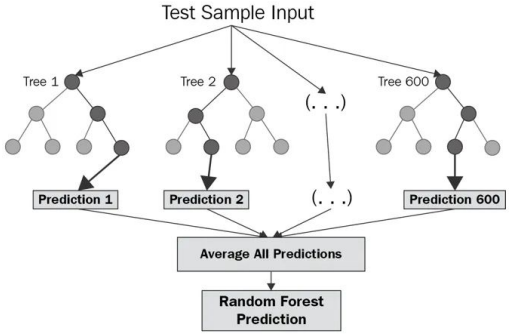
- A single decision tree tends to overfit the training data and perform poorly on test data
- To reduce variance, ensemble methods can be used
- Random Forest uses **bagging**: individual decision trees are trained in a parallel way, and each tree is trained by a random subset of the data



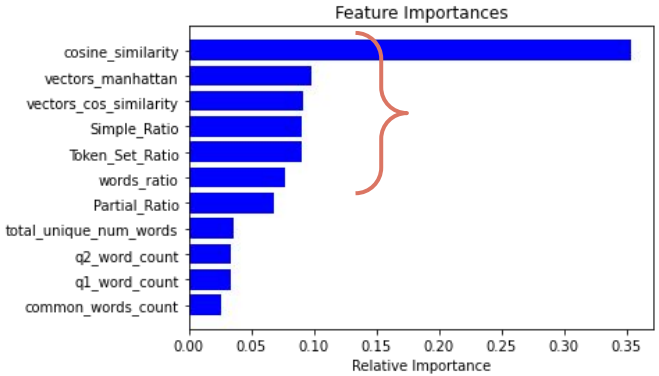
RANDOM FOREST

- **Hyperparameters:**
  - ➢ Bootstrap = True; bootstrap aggregation can be used to reduce variance
  - ➢ Maximum number of features considered for splitting node = 'auto' (or sqrt(no. features))
  - ➢ Minimum number of data allowed in leaf node = 2
  - ➢ Minimum number of data in node before node is split = 2
  - ➢ Number of trees in forest = 1062
- Hyperparameters were determined using cross validation (RandomizedSearchCV and GridSearchCV)

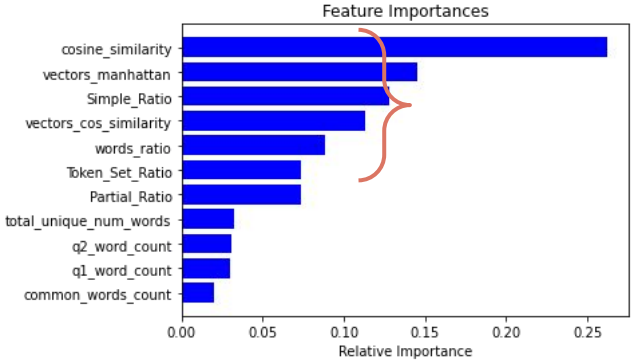




RANDOM FOREST



cleaned_features



stopwords_lemmatize

- Common patterns: cosine_similarity is the major deciding factor for Random Forest
- Token_Set_Ratio, Simple_Ratio, words_ratio, vector_manhanttan and vector_cos_similarity follow
- Word count related features have low importance
- We pick the above 6 features to train the model

RANDOM FOREST

cleaned_features
Accuracy: 0.824000 Precision: 0.740210 Recall: 0.800655 F1 Score: 0.769247 AUC-ROC: 0.819078 Log Loss: 0.370518

stopwords_lemmatize
Accuracy: 0.778520 Precision: 0.686196 Recall: 0.728821 F1 Score: 0.706866 AUC-ROC: 0.768041 Log Loss: 0.432070

- cleaned_features gives the best metrics compared to the other dataset
- Lemmatization may have introduced noises, and thus performances of this dataset are not as good as the 1st one
- e.g: Row index 3
  - Original text: why am i mentally very lonely how can i solve it
  - Lemmatized: mentally lonely solve

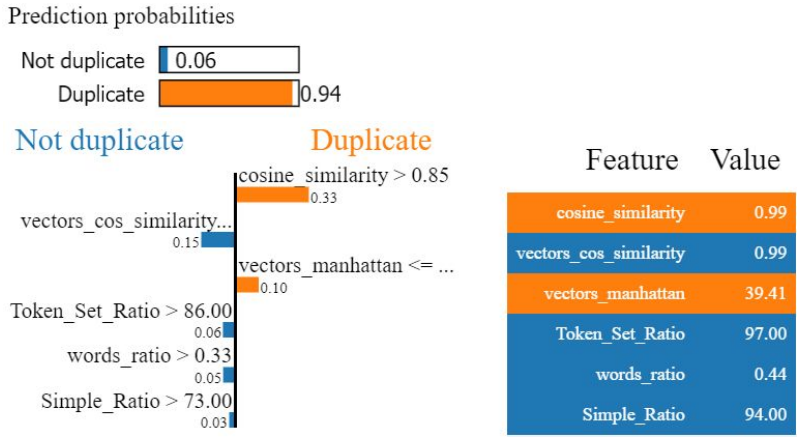
RANDOM FOREST

Correct prediction: true positive

Actual: Duplicate
Predicted: Duplicate

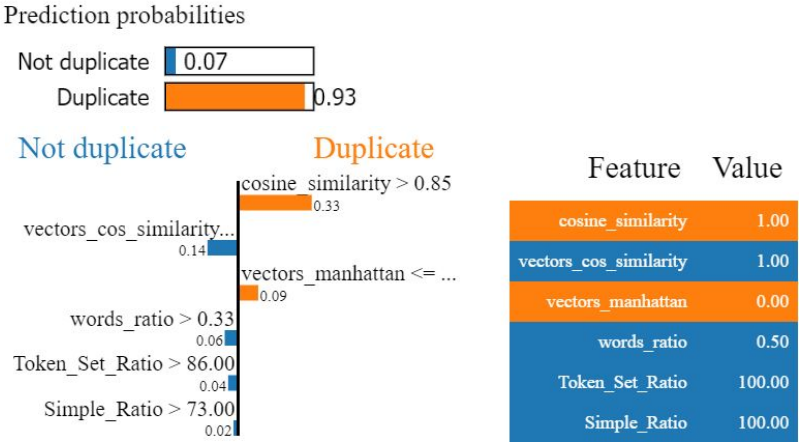
cleaned_features

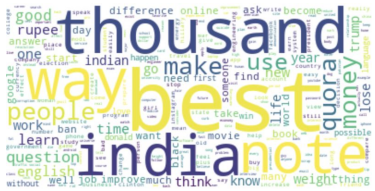
Question 1: “is shopnix a better ecommerce portal or shopify”
Question 2: “is shopnix a better ecommerce portal than shopify”



stopwords_lemmatize

Question 1: “shopnix well ecommerce portal shopify”
Question 2: “shopnix well ecommerce portal shopify”





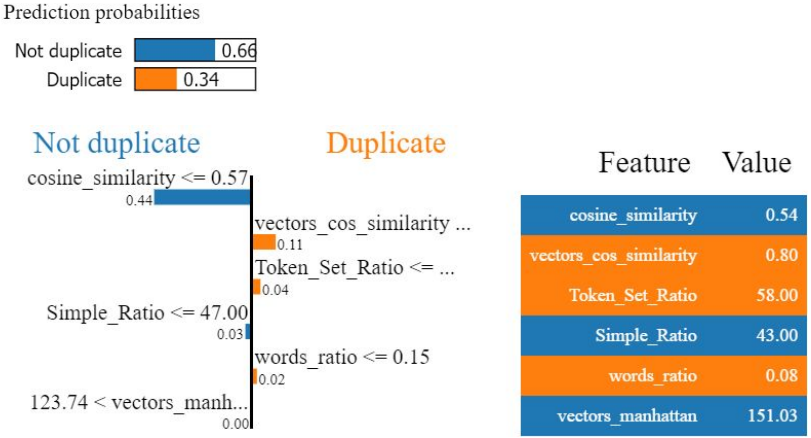
RANDOM FOREST

Incorrect prediction: false negative

Actual: Duplicate
Predicted: Not duplicate

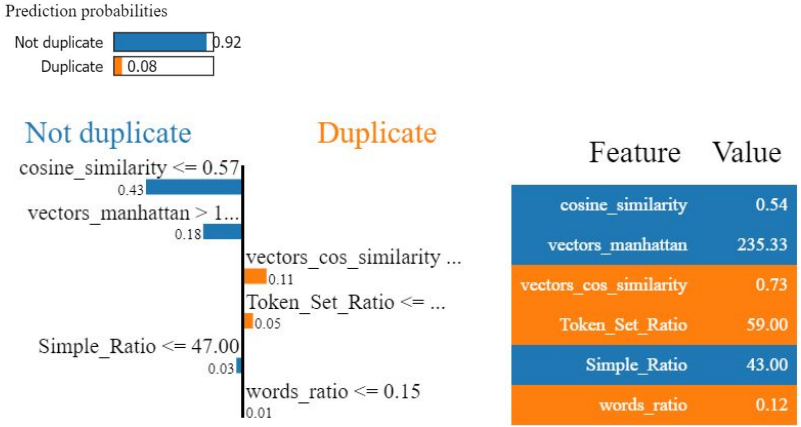
cleaned_features

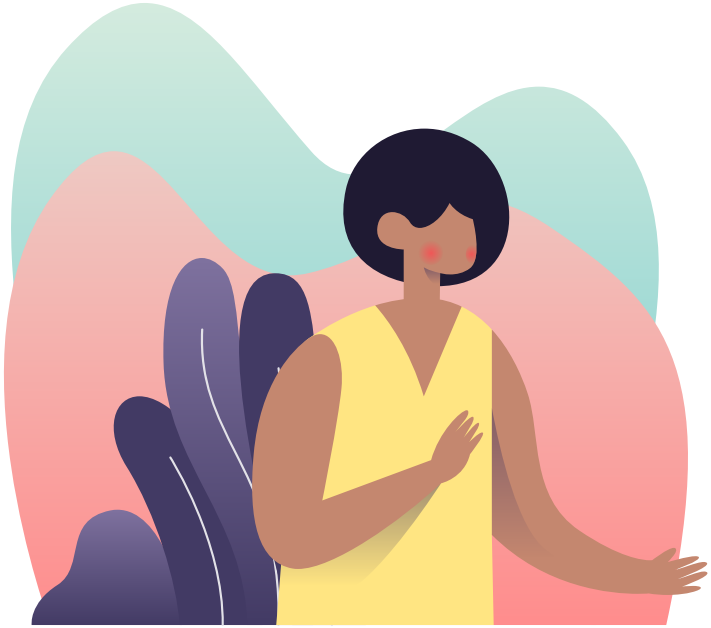
Question 1: "what would be the scenario for indian students after donald trump becomes the president of us"
Question 2: "how will trump s presidency affect prospective international students from syria"



stopwords_lemmatize

Question 1: "would scenario indian student donald trump becomes president u"
Question 2: "trump presidency affect prospective international student syria"





04

EVALUATION

MACROSCOPIC PERFORMANCE

Performance metrics	
Accuracy	Ratio of correct predictions to total instances
Precision	Maximise if false positives are expensive
Recall	Maximise if false negatives are expensive
F1 Score	Harmonic mean of precision and recall
AUC-ROC	Ability of model to separate positive and negative classes
Log-Loss	Penalises false classifications

MACROSCOPIC PERFORMANCE

Models	Logistic Regression	LSTM	Random Forest
Accuracy	0.761	0.853	0.879
Precision	0.730	0.835	0.834
Recall	0.830	0.879	0.834
F1 Score	0.761	0.853	0.834
AUC-ROC	0.762	0.935	0.869
Log-Loss	0.484	0.357	0.272

MICROSCOPIC PERFORMANCE

Cleaned_features

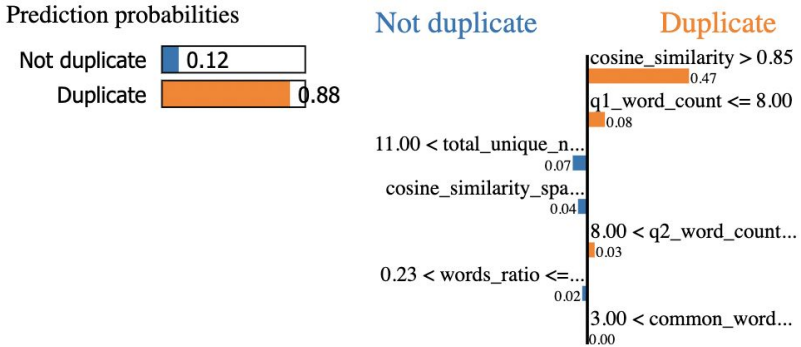
Actual: is duplicate

Predicted: is duplicate

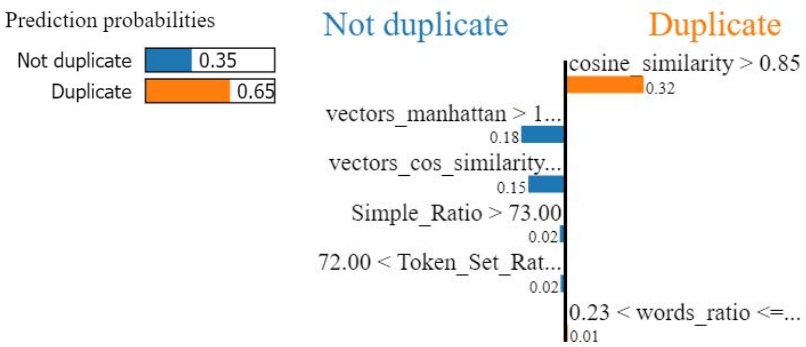
Question 1: “how are you protecting yourself from identity theft”

Question 2: “what are good ways to protect yourself from identity theft”

Logistic Regression



Random Forest



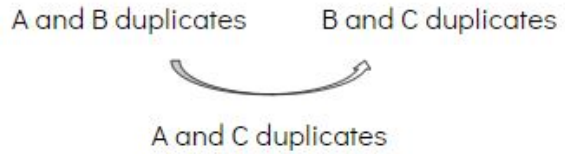
05

CONCLUSION

FUTURE IMPROVEMENTS

1

Pre-processing Step



- Mislabelled data
  - ➢ Labels in train set were provided by experts and may be subjective
  - ➢ In our project, we assumed the labels were ground truth

Example:
Question 1: “How can I increase my height after 21 also”
Question 2: “Can height increase after 25?”
Training labels of duplicated sentences

- Tackling class imbalance
  - ➢ Data augmentation by replacing adjectives and verbs by synonyms to generate new text data

- Other distance metrics
  - ➢ Jaccard Similarity: repeated words in sentences do not matter

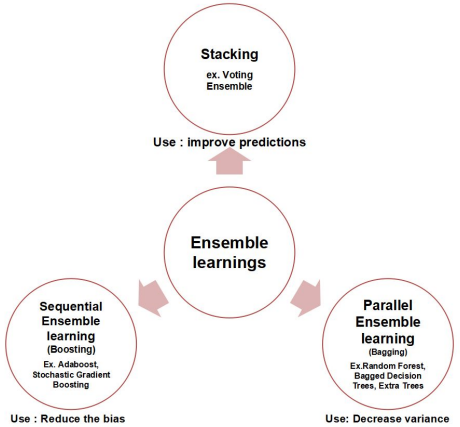
FUTURE IMPROVEMENTS

2

Modelling Step

- Incorporate frameworks in LSTM model
  - ➢ “Compare-aggregate” framework: word-level matching followed by aggregation using Convolutional Neural Networks
  - ➢ “Attention-based” framework: to highlight relevant features of the input data

- Build ensemble models that comprise all models we have built



Thank you!